Project Proposal Summary Sheet

Project Number	P_2024_03
Project name	*P_2024_03_ASAM ARTI_V2_F*
Domain	Measurement & Calibration
Impacted standard(s)	ASAM Run-Time Interface (ARTI)
Project type	StandardDevelopment – Future Version Development
Start date	01.10.2024
End date	30.11.2025
TSC Submission:	14.06.2024
Proposer(s)	Matthias Scheid, Tasking Marco Eichenseer, Vector Thomas Gröger, PLS Rudi Dienstbeck, Lauterbach
ASAM Office Responsible (OR)	Yash Shah
Initiating Companies	Tasking, Vector, PLS, Lauterbach
ASAM funds	
Backward Compatibility	No. The proposal introduces breaking changes.

For more information on the ASAM project process and the proposal phase in particular, please refer to the ASAM Project Guide.

Table of Contents

1 Executive Summary

The proposal will introduce a new version of the ASAM Run-Time Interface Base Standard Version 1.0.0. The standard includes a standardized interface data model and file format for the collaborating tools with special focus on collecting run-time information over a certain period of time (aka "tracing").

The ASAM ARTI standard closely maps to the AUTOSAR ARTI standard. The AUTOSAR ARTI standard was extended in the last years without reflecting the changes in ASAM ARTI. This proposal's purpose is to align the new features of the AUTOSAR ARTI standard, to fix inconsistencies identified in the existing standard and to improve the wording.

The outcome of this proposal will be a new version of the ASAM ARTI standard. It will improve the interoperability of tools exchanging trace information, like tracing tools and timing analysis tools.

2 Overview / Goals

2.1 Motivation

The ASAM ARTI standard links to the AUTOSAR ARTI standard. The AUTOSAR ARTI standard was extended with each new AUTOSAR release. ASAM ARTI should be aligned with the latest AUTOSAR ARTI release (R23-11).

While ASAM ARTI relates to AUTOSAR ARTI, it should stand for itself without the need of this relationship. Some wordings in the existing v1.0.0 version rely on AUTOSAR terminology. It is the intention of the new version to remove AUTOSAR terminology as ASAM ARTI is also used in non-AUTOSAR projects.

While using the ASAM ARTI standard v1.0.0 in real projects, we identified inconsistencies and missing items. The goal of this proposal is also to eliminate these inconsistencies and add the missing information.

These points mentioned above encompass minor wording changes, backward compatible additional, and breaking changes to existing Trace Classes.
Proposing a new major version is mainly motivated by the fact that data encoded following the new specification will not be compatible anymore. While the introduction of a new Trace Class or an additional Channel in all Trace Classes would be backward compatible, changing the name, order, or content of existing channels would be a breaking change. Such a breaking change is considered necessary because:

- core information is not considered correctly in all classes that need it in v1.0.0
- the “eventParameter” of the trace class SERIVCECALLS needs adjustment for further servicecalls.
- The events of the trace class SPINLOCK need to be adjusted to match the AUTOSAR definition

2.2 Relations to Other Standards, Projects, or Organizations

Standard and Standardization activities

This standard relates to the AUTOSAR standard “AUTOSAR Run-Time Interface” in the version R23-11 (AUTOSAR_CP_SWS_ARTI.pdf and AUTOSAR_CP_SWS_OS.pdf).

Backward Compatibility to earlier releases

The new version will include breaking changes of the data format. This means, it is not backward compatible. Thus, a new major version is introduced.

3 Technical Content

3.1 General

Scope of the proposed changes

The changes proposed in this document include minor wording changes, backward-compatible additions, and breaking changes.

Clarification of data types

The “type” column of Trace Classes is specified as "Token, literal text" or "uint32" (but with conversion rule). It should be clarified how this name mapping works and when a conversion rule is required.

Example:
The "eventName" channel for AR_CP_OS_TASK has the type "Token, literal Text" and contains values like "OSTask_Start".

The "eventParameter" channel has the type uint32 but is expected to be mapped to actual name via conversion rule.

Improve wording for optional or not available information.

It should be clarified what value a channel contains when the value is missing.

Example:
The "instanceName" channel should contain the "OS Short Name" but is always set to "0" in the examples seen.
Note: Neither tracing nor analysis tools currently use this information.

Handling of "ASAM Specification Version" in MDF

So far there seems to be no "ASAM Standard Version" info in the MDF.

This should be addressed when another specification version is published to distinguish between versions when reading the file.

Note: The metadata information "Tool Version" and "File Version" mean something different.

Improve use of AUTOSAR wording

Parts of the ASAM specification heavily rely on the wording used by AUTOSAR (e.g. "OS Short Name") and the start of the Trace Classes chapter contains the ARTI_TRACE macro.

This leads to discrepancies where the wording is not applicable, or descriptions of Trace Classes do not match how said macro is defined in ARTI.

Data types do not follow a well-established wording scheme. AUTOSAR types would be all upper case (e.g. UINT32), a C++ fixed size integer would be uint32_t, the document currently uses uint32.

Handling of Core identifier

Several trace classes need a core identifier to transport the CPU core the event originated from. The trace class AR_CP_OS_CAT2ISR for example uses the “instanceParameter” channel for this information.

This is also the case for the trace class AR_CP_RTE_RUNNABLE, this is however not consistent with the underlying AUTOSAR macro.

This seems also to be the case for trace class AR_CP_OS_SPINLOCK (note the defect in this class described in the following chapter).

It should be clarified which trace classes need this information and how to transport it. It has been briefly discussed that the “instanceParameter” channel cannot contain the core identifier in all cases. An alternative to avoid inconsistencies would be a dedicated core identifier channel.

3.2 Improvements to Trace Classes

AR_CP_OS_CAT1ISR

A trace class for Category 1 ISRs should be added. It will most likely be an exact duplicate of CAT2ISR except for the name.

Note: Adding additional trace classes would be a backwards compatible change

AR_CP_OS_SPINLOCK

The "instanceParameter" channel is described as "Not used, should be set to 0."

It is, however, used (in examples and expected by our tooling) to transport a "Core identifier" (as in CAT2ISR e.g.). This has already been discussed briefly and has been identified as a defect.

AR_CP_OS_SERVICECALLS

The problem described for the "instanceParameter" channel of Trace class AR_CP_OS_SPINLOCK also applies to SERVICECALLS.

The "eventName" channel description is not complete. It does not reflect all events defined by AUTOSAR. The trace class does not specify entries and exits which are necessary for correct nesting.

The "eventParameter" channel is described as "Either nesting depth or the Spinlock identifier". The SPINLOCK trace class uses a conversion rule which allows the spinlock name to be resolved but the SERVICECALLS trace class does not use a conversion rule in practice.

We currently resolve using conversion rule on spinlock channel. The reason for not setting a conversion rule seems to be to prevent accidentally resolving nesting depth as spinlock names. It should be clarified how entity names can be resolved in such situations.

AR_CP_RTE_RUNNABLE

The "instanceName" channel is described to contain "Value that identifies the component type." In the examples we have, this channel always contains the string "My Operating System". This has already been discussed briefly. This information may not be available.

The "instanceParameter" channel is described as "Core identifier" which does not match the AUTOSAR macro.

AR_CP_SCHM_SCHEDULABLE

The problem described for the "instanceName" channel of Trace class AR_CP_RTE_RUNNABLE also applies to SCHM_SCHEDULABLE.

USER_STOPWATCH

We haven't worked with USER_STOPWATCH events yet.
So far, no changes to the USER_STOPATCH trace class are proposed but the content of this trace class should be checked during this update.

USER_DATAFLOW_STOPWATCH

We haven't worked with USER_STOPWATCH events yet.
So far, no changes to the USER_STOPATCH trace class are proposed but the content of this trace class should be checked during this update.

USER_DATAPOINT

The "instanceName" is described as "Value that identifies the instance of the dataflow stopwatch". This has already been discussed briefly and is indeed a defect. The description should refer to the “datapoint”, not the “stopwatch”.

The "eventParameter" channel is specified to be uint32 but contains a value of the type specified in the "eventName" channel (bool, uint16, float, etc...). The binary representation does not seem to be specified. It should be clarified what the exact binary representation of the possible data types is when stored as uint32.

Example:
How is the float value 1.25f stored exactly in the MDF file? Is it guaranteed to be equal to "00111111101000000000000000000000"(BE)?
How is the signed int8 value -120 stored example in the MDF file? Is it guaranteed to be equal to "11111111111111111111111110001000"(BE) or should the upper 24 bit be filled with zeros like this "00000000000000000000000010001000" (BE)?

4 Deliverables

Table 1 Deliverables

Item No.	Description
1	ASAM Run-Time Interface Specification
2	ASAM ARTI Example File

4.1 Review Process

Table 2 Selection of Review Type

Please indicate whether the project is aiming to perform an ASAM member review or a full public review. This is not required for maintenance projects.	ASAM Member Review

5 References

[1] ASAM; ASAM Run-Time Interface; 2020; https://www.asam.net/standards/detail/arti/

[2] AUTOSAR; AUTOSAR Run-Time Interface; 2023; https://www.autosar.org/fileadmin/standards/R23-11/CP/AUTOSAR_CP_SWS_ARTI.pdf

[3] AUTOSAR; Specification of Operating System; 2023; https://autosar.org/fileadmin/standards/R23-11/CP/AUTOSAR_CP_SWS_OS.pdf